|  |  |  |  |
| --- | --- | --- | --- |
| **Responsive Website** | | | |
| **S.NO** | **Command/Problem** | **Description 1** | **Description 2** |
|  | Npm run dev is used to start server | Write this in CLI |  |
|  | Font:inherit command in css | It reset all fonts in site, or inherit characters from its parent elements |  |
|  | How to pick color from website using picker ? | Goto website and right click and goto inspect than go to desired color on website, go to style and go to color icon in style and use picker from color symbol than use hsl or rgb format, copt it and paste in your style |  |
|  | **--clr-primary-400 command line means in css style ?** | **--clr-primary-400** might represent the 400-level shade of your primary color. This could be a slightly darker or lighter variation compared to the base primary color (**--clr-primary**), which might be defined elsewhere in your CSS. |  |
|  | How to convert Px for font in to rem ? | Divide the font size in pixel by 16, you will get rem |  |
|  | Ctrl + D to delete same words from multiples lines at same time |  |  |
|  | <ul role="list" class="nav-list"> mean in html ? | User can interpret role easily |  |
|  | How to insert mail in form and click side button ? | <div>  **<form action="">**              <button class="button"> Go </button>              <p> Copyright 2024. All Rights Reserved. </p>              </form>  </div> | |
|  | width: min(var(--max-width, 100%)); | Choose a min width btw max width or 100pc in style |  |
|  | To auto both margins left right | margin-inline: auto; |  |
|  | To make small space at left margin so that words don’t get touched | Use in style  width: min(var(--max-width), 100% - var(--padding));  width: calc(100% - (var(--padding) \* **3**));  3 is rem |  |
|  | even-columns { display: grid; gap: 1rem; | 1. **.even-columns**: This is a class selector targeting elements with the class name **even-columns**. 2. **display: grid;**: This property is setting the display mode of the selected elements to be a grid layout. Grid layout allows for a two-dimensional layout system, where elements can be positioned in rows and columns. 3. **gap: 1rem;**: This property sets the gap, or the spacing between grid items, to 1rem (1 rem is equivalent to the font size of the root element). This creates space between the grid items to visually separate them and make the layout more readable or aesthetically pleasing. | |
|  | In style.css  @media (min-width: 50em) {  .even-columns {  grid-auto-flow: column;  grid-auto-columns: 1fr;  }  } | This CSS rule is a media query targeting devices with a minimum width of 50em (which is commonly used to target larger screens such as tablets and desktops). Within this media query, there are styles applied to elements with the class **.even-columns**. Let's break down what each line means:   1. **@media (min-width: 50em) { ... }**: This is a media query that specifies styles to be applied only when the viewport width is at least 50em. 2. **.even-columns { ... }**: This targets elements with the class **.even-columns** within the specified media query. 3. **grid-auto-flow: column;**: This property sets the direction in which grid items are automatically placed within the grid contfainer. In this case, it's setting them to flow into columns, meaning they will be placed in a single column, filling each column before moving to the next one. 4. **grid-auto-columns: 1fr;**: This property sets the size of the implicitly-created columns in the grid. In this case, it's setting each column to take up an equal portion of the available space within the grid container. The **fr** unit stands for "fraction unit," and **1fr** means each column will take up an equal fraction of the available space.   So, in summary, within this media query, elements with the class **.even-columns** will be displayed as a single-column grid layout, with each column taking up an equal portion of the available space. This layout adjustment is triggered when the viewport width is at least 50em, typically targeting larger screens. | |
|  | In css style how to put radius on border of button plus padding ? | .button {    border-radius: 100vh;    padding: 1em 2em;    font-weight: var(--fw-bold);    border: 0;    font-size: var(--fs-button);    color: var(--clr-neutral-100);    background-color: var(--clr-accent-400);  } |  |
|  | In css style  .button:hover,  .button:focus-visible {  background-color: var(--clr-accent-300);  } | Hover when u move cursor over button  Cursor is when u click on button , in both cases it changes the state  So, in summary, **.button:hover** changes the background color of elements with the class **.button** when the mouse is hovering over them, while **.button:focus-visible** changes the background color when those elements receive focus and the focus is visibly indicated to the user. This provides visual feedback to users during interaction with the elements.  .button[data-type="inverted"]:**hover**,  .button[data-type="inverted"]:**focus-visible** {    background-color: var(--clr-neutral-100);    color: var(--clr-accent-300);  }  Top of Form | |
|  | Padding inside button function | padding: 1.25em 2.5em;  this 1.25 is vertically increasing button from text inside and other is horizontally | |
|  | Span element in burron | In the provided HTML snippet, the **<span>** element is used to wrap the text "Menu" inside the **<button>** element.  <**button** class="mobile-nav-toggle" aria-control="primary-navigation">  <span> Menue </span>  </**button**> | |
|  | If you want one icon on top left side to get appeared ? | Make a class of button  U should know the class of image whom you want to disappear  .mobile-nav-toggle .icon-close {    display: none;  } |  |
|  | To remove border of icon | Use this command inside function border:0; |  |
|  | To make background of icon transparent | Use this command inside function border:0; |  |
|  | To remove button for time being, what is command ? | <**button** class="button" **style="display:none**"> Get Started </**button**> |  |
|  | How we put flex in header ?  AT first the items were up and down in header like logo and menue  How to put them in center and than justify them? | * 1. Put this function in style.css   .nav-wrapper {  display: flex;  align-items: center;}   * 1. We have used **align-items: center**;} to make all icons or items left to right and at ccenter   2. In html, put div after container   <header class = "primary-header">  <div class = "container">  **<div class = "nav-wrapper">**   * 1. **To justify use command justify-content: space between, in this way the icons will move to their lcations at left right with space**   **.nav-wrapper {**  **display: flex;**  **align-items: center;**  **justify-content: space-between;**  **}** | |
|  | To remove menu and add only block on header ? | Make a function of media which will display none of items in header such as  media (max-width: 50em) {    .primary-navigation {    display: none;    }  }  Now add all mobile vav funtions that include items in style.csss inside this media function and tell him to display atleast the block  media (max-width: 50em) {  .primary-navigation {  display: none;  }  .mobile-nav-toggle {  display: block;  cursor: pointer;  padding: 0.5em;  background: transparent;  border:0;  } | |
|  | The code in main.js means ?  const navToggle = document.querySelector(".mobile-nav-toggle");  const primaryNav = document.querySelector(".primary-navigation"); | This line of code is written in JavaScript and it selects the first element in the document that has a class of "mobile-nav-toggle" using the querySelector method. | |
|  | navToggle.addEventListener("click", () => {  primaryNav.toggleAttribute("data-visible");  } ) | The line **primaryNav.toggleAttribute("data-visible");** toggles the presence of the HTML attribute **data-visible** on the **primaryNav** element.  This technique is often used to control the visibility or behavior of elements using custom data attributes. In this case, toggling the **data-visible** attribute might be part of a JavaScript function responsible for showing or hiding the navigation menu based on certain conditions. | |
|  | I want to click on humbergur icon and it should display menu along with going to console it tells u status  Area expanded shows status when true or false | Make a code on main.js as  const navToggle = document.querySelector(".mobile-nav-toggle");  const primaryNav = document.querySelector(".primary-navigation");  navToggle.addEventListener("click", () => {  **primaryNav.hasAttribute("data-visible")**  **? console.log("true")**  **: console.log("it doesnt");**        primaryNav.toggleAttribute("data-visible");  } );  Go to index.html and add this code  <header class = "primary-header">        <div class = "container">          <div class = "nav-wrapper">            <a href="#"><img src="images/logo.svg" alt="manage"> </a>            <button class="mobile-nav-toggle" **aria-controls="primary-navigation"** **aria-expanded="false"**>            <img class="icon-close" src="images/icon-close.svg" alt="" aria-hidden="true">            <img class="icon-hamburger" src="images/icon-hamburger.svg" alt="" aria-hidden="true">            <span class="visually-hidden"> Menu </span></button>            <nav class = "primary-navigation" id="primary-navigation">              <ul aria-label="Primary" role="list" class="nav-list"> | |
|  | To see the status in aria-expanded by going to inspect in element “index.html” | const navToggle = document.querySelector(".mobile-nav-toggle");  const primaryNav = document.querySelector(".primary-navigation");  navToggle.addEventListener("click", () => {  **primaryNav.hasAttribute("data-visible")**  **? navToggle.setAttribute("aria-expanded", false)**  **: navToggle.setAttribute("aria-expanded", true);**        primaryNav.toggleAttribute("data-visible");  } ); | |
|  | To make gap between menu items | Add display grid than put gap  .nav-list {      display: grid;      gap: var(--size-100);    } | |
|  | Some common pseudo-elements include **::before** and **::after** | which allow you to insert content before or after the content of an element, respectively, and style it separately.  For example, you can use **::before** to insert an element before the content of a paragraph and style it to act as a decorative element such as a bullet point or a decorative border.  **p::before** {  content: "•";  color: red;  }  .**primary-navigation::after {**      content: "";      position: fixed;      z-index: -1;      inset: 0;      background-image: linear-gradient(red, blue);        }  **z-index: -1;**: This property sets the stacking order of the pseudo-element behind the primary navigation, ensuring that it stays behind other content on the page.  **inset: 0;**: It sets the top, right, bottom, and left positions of the pseudo-element to 0, making it cover the entire viewport.  **background-image: linear-gradient(red, blue);**: This property creates a linear gradient background for the pseudo-element, transitioning from red to blue horizontally.  **position: fixed;**: It fixes the position of the pseudo-element relative to the viewport. | |
|  | To put sides of menu colored with gradient……Instead of doing the gradient test on nav data, do it on main header class as primary class | Suppose the header class is “primary-header”  Go to main.js and add const primaryHeader to take data from primary-header class such as:  const primaryHeader = document.querySelector(".primary-header");  Add primary-header line below the event function to toggle data-overlay of primary header class  **const primaryHeader = document.querySelector(".primary-header");**  const navToggle = document.querySelector(".mobile-nav-toggle");  const primaryNav = document.querySelector(".primary-navigation");    navToggle.addEventListener("click", () => {      primaryNav.hasAttribute("data-visible")          ? navToggle.setAttribute("aria-expanded", true)          : navToggle.setAttribute("aria-expanded", false);        primaryNav.toggleAttribute("data-visible");  **primaryHeader.toggleAttribute("data-overlay")**  } );  Go to style.css and add primary header[data-layover] class in media with max width: 50 em  @media (max-width: 50em) {    .primary-navigation {      display: none;      padding: var(--size-600);      position: absolute;      inset: 7rem var(--size-400) auto;      background: var(--clr-neutral-100);      border-radius: var(--size-100);      box-shadow: 0 0 0 0.75em rgb(0, 0, 0, 0.05);      }  **.primary-header[data-overlay] {**  **content: "";**  **position: absolute;**  **inset: 0;**  **background-image: linear-gradient(red, blue);**  **}** | |
|  | If after implementation of gradient, the top margin is somehow in header is white instead of blue red ? | Go to padding in primary heading and adjust variable size | |
|  | To make linear gradient transparent instead of red blue  background-image: linear-gradient (rgb(0, 0, 0 / 0)), rgb(0, 0, 0 / 0)); | In your example, it seems like there's a syntax error with the **linear-gradient()** function. There's an extra closing parenthesis and a comma. Also, the colors specified are fully transparent (**rgb(0, 0, 0 / 0)**), which means the gradient will be invisible. | |
|  | To move hamburger icon anywhere u want | .mobile-nav-toggle {      display: block;      position: fixed;  **top: var(--size-600);**  **right: var(--size-400);** | |
|  | Always put utility class below of style.css, it damage the order |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  | When menu (primary-navigation) opens, it displays background black color, also put some border radius | osition: fixed;      inset: 7rem var(--size-400) auto;      background-color: var(--clr-neutral-900);      border-radius: var(--size-100);      gap: var()      }  The **inset** property is a shorthand for setting the top, right, bottom, and left offsets of an absolutely or fixed positioned element. In this case:   * **7rem** sets the top offset to **7rem**. * **var(--size-400)** sets the right offset to the value of the CSS variable **--size-400**. * **auto** sets the bottom offset to **auto**, meaning it will be automatically determined by the browser. | |
|  | To remove underlines from menu | text-decoration: none; | |
|  | I don’t want my menu to go bottom, it should remain at its place | position: absolute; | |
|  | To put box shadow around border | box-shadow: 0 0 0 0.75em rgb(0, 0, 0, 0.05); | |
|  | How to make background color mix like red and blue other than nav menu | The CSS you provided targets the pseudo-element **::after** of the **.primary-navigation** class. It creates a fixed-positioned pseudo-element behind the primary navigation menu with a linear gradient background from red to blue covering the entire viewport.  .primary-navigation::after {      content: "";      position: fixed;      z-index: -1;      inset: 0;      background-image: linear-gradient(red, blue);        } | |
|  | In order to put menu (nav-list) in right to left order using flex | .nav-list {    display: flex;  }  In order to put gap left to right  .nav-list {    display: flex;    gap: var(--size-400)  } | |
|  | gap: clamp(var(--size-500), 5vw, var(--size-800)); | 1. The minimum value (**var(--size-500)**): The minimum allowed value for the gap size. 2. The preferred value (**5vw**): The preferred or ideal value for the gap size. 3. The maximum value (**var(--size-800)**): The maximum allowed value for the gap size.   So, the **gap** property in your example will result in a gap size that is responsive and varies between **var(--size-500)** and **var(--size-800)**, with a preference for **5vw** when the viewport width allows it.  Top of Form | |
|  | To stick menu nav box to left side and broader or shorter its width ? | @media (max-width: 50em) {    .primary-navigation {      display: none;      padding: var(--size-700);  **max-width: 25rem;**  **margin-left: auto**;      position: fixed;      inset: 7rem var(--size-400) auto;      background: var(--clr-neutral-100);      border-radius: var(--size-100);      box-shadow: 0 0 0 0.75em rgb(0, 0, 0, 0.05);      } | |
|  | How to select elements adjacent and how to make outline ?  .flow \* + \* {  outline: 3px solid lime;  } | So, **.flow \* + \*** selects any element that is a sibling (immediate neighbor) of another element within a parent with the class **.flow**. | |
|  | Flow is used to specify the point in html and than provide space  <div class="container">          <div class="even-columns">  **<div class="flow">**            <div>abcd</div> | :where(.flow :not(:first-child)) {    margin-top: 1em;  } | |
|  | In order to put line spaces between <Li>  Use div container method across each <li> | <li>  **<div class="flow" style="--flow-spacer: 1.5em">**                    <h3 class="fs-600 fw-bold"> Track company wide progress                    </h3>                    <p> · Outfit Font Family is an eye-catching sans serif font designed by font designer Rodrigo cda for use by automation specialists at outfit.</p>  **</div>**                </li>                <li>  **<div class="flow" style="--flow-spacer: 1.5em">**                    <h3 class="fs-600 fw-bold"> Advanced built in Reports </h3>                    <p> · Outfit Font Family is an eye-catching sans serif font designed by font designer Rodrigo Fuenzalida for use by automation specialists at outfit.</p>  **</div>**                </li>                <li>                  <div class="flow" style="--flow-spacer: 1.5em">                    <h3 class="fs-600 fw-bold"> Everything you need in one place </h3>                    <p> · Outfit Font Family is an eye-catching sans serif font designed by font designer Rodrigo Fuenzalida for use by automation specialists at outfit.</p>                  </div>                </li> | |
|  | To make visibility of all <p> | p {  **opacity: 0.7;**  } | |
|  | To shorten the “p” or para width | p {    opacity: 0.7;  **max-width: 35ch;**  }  **35ch**: This value sets the maximum width to be 35 times the width of the "0" character in the element's font. This means the maximum width of the element will be equivalent to 35 characters in the given font. | |
|  | If I want not to target heading | p:**not([class])** {    opacity: 0.1;    max-width: 35ch;  }  This CSS rule targets **<p>** elements that do not have a **class** attribute and applies the following styles to them: | |
|  | If you want the width of your para to 100 pc? Using same p class? | Make a new class with p name along with data-width=wise    p**[data-width="wide"]** {    max-width: 100%;  }  In html add on each p as  **<p data-width="wide">** | |
|  | How to move button to right side ?  On “display flex=”inline”, you cannot move button on right side  On flex, you cannot apply justify-self-end | First make div with class=”justify-self-end” after the button command in html  <**div class="justify-self-end"** >  <button class="button | push-right" data-type="inverted"> Get Started </button>  </div>  2ndly, make a utility function of justify-self-end like this  .justify-self-end {  **justify-self: end;**  } | |
|  | How to align items at center ? specially the button | Make style.css as  .vertical-align-center {    align-items: center;  }  In html,  div class="container">  **<div class="even-columns vertical-align-center">**            <div>              <p class="fs-secondary-heading fw-bold"> Simplify!! How your team is working </p>            </div>              <div class="justify-self-end" > **<button class="button** | push-right" data-type="inverted"> Get Started </button>  </div> | |
|  | Form was not showing next to button due to lack of input type at footer side | <div>  <form action="">  **<input type="email">**  <button class="button">Go</button>  <p>Copyright 2020. All Rights Reserved</p>  </form>  </div> | |
|  | How to make background color red in CTA? | If u don’t make a function of cta with position relative the background will not change  .**cta { position: relative}**  .cta::before {    content: "";    position: absolute;    inset: 0;  **background: red;**    background-image: url(      "images/"    ); | |
|  | If you want to work witin entire section not by going outside like while applying color | .cta {    position: relative;  **isolation: isolate;**  }  The isolation property in CSS controls how an element's content is isolated from its surroundings. When you set isolation: isolate;, it creates a new stacking context for the element, which means that the element's content will be rendered in its own layer separate from its parent and sibling elements. | |
|  | U can also reduce the visibility of background image bubble using opacity on cta section | .cta {    position: relative;    isolation: isolate;  }  .cta::before {    content: "";    position: absolute;    inset: 0;    z-index: -1;    background: red;    background-image: url("images/bg-tablet-pattern.svg");  **opacity: 0.2 ;**  } | |
|  | To remove the repetition of background images | .cta::before {    content: "";    position: absolute;    inset: 0;    z-index: -1;    background: red;    background-image: url("images/bg-tablet-pattern.svg");    opacity: 0.1 ;    **background-repeat: no-repeat;**  } | |
|  | To change background position ? | .cta {    position: relative;    isolation: isolate;  }  .cta::before {    content: "";    position: absolute;    inset: 0;    z-index: -1;    background: red;    background-image: url("images/bg-tablet-pattern.svg");    opacity: 0.1 ;    background-repeat: no-repeat;  **background-position: 0 100%;**  }  Here's what the values **-20rem -5rem** represent:   * **-20rem**: This specifies the horizontal position of the background image. In this case, it's set to **-20rem**, which means the background image will be shifted 20 *rem* (root em) units to the left of the container's left edge. * **-5rem**: This specifies the vertical position of the background image. It's set to **-5rem**, which means the background image will be shifted 5 *rem* units upwards from the container's top edge. | |
|  | To move button to center | <div class**="justify-self-end-md"** > <button class="button | push-right" data-type="inverted"> Get Started </button>  </div> | |
|  | Body or p don’t have .function in style.css |  | |
|  | To change nav of footer and to put hover and focus ? | .**footer-nav a** {    color: var(--clr-neutral-100);    text-decoration: none;  }  .**footer-nav a:is(:hover, :focus)** {    color: var(--clr-accent-400);    } | |
|  | To divide 1 colum of footer-nav in to 2 column | .footer-nav {  **columns: 2**;  } | |
|  | How to add svg links in visual studio ? as I want to add pictures in vertical position, but I don’t have their codes to place them in footer | First make a new file in images folder with name social-icons.svg  Now go to website of svgsprite and add all picture of svg there  You will get codes of all links  Copy the codes and paste in **social-icons.svg** file  Also implement the code in htmls as  <ul role="list" aria-label="Social links">                  <li> <a aria-label="facebook" href="#"> </a>  **<svg class="icon">**  **<use xlink:href="images/social-icons.svg#icon-facebook">**</use>  **</svg>**                </li>                <li> <a aria-label="Youtube" href="#"> </a>  **<svg class="icon">**  **<use xlink:href="images/social-icons.svg#icon-youtube">**</use>  **</svg>**                </li>                <li> <a aria-label="Twitter" href="#"> </a>                <svg class="icon">                  <use xlink:href="images/social-icons.svg#icon-twitter"></use>                </svg>                </li>                <li> <a aria-label="pinterest" href="#"> </a>                <svg class="icon">                  <use xlink:href="images/social-icons.svg#icon-pinterest"></use>                </svg>                </li>                <li> <a aria-label="Instagram" href="#"> </a>                <svg class="icon">                  <use xlink:href="images/social-icons.svg#icon-instagram"></use>                </svg>                </li>                </ul> | |
|  | To put all icons in horizontal position, use flex and make class in UL of nav-list for icns | .social-list {  **display: flex;**  }  <ul **class="social-list"** role="list" aria-label="Social links">                  <li> <a aria-label="facebook" href="#"> </a>  **<svg class="social-icon">**  **<use xlink:href="images/social-icons.svg#icon-facebook"></use>**  **</svg>**                </li>  Make class of each svg to social-icon | |
|  | To reduce the size of each icons ? or width of icon in footer like insta,fb | .social-icon {  **width: var(--size-600);**  } | |
|  | How to change color of icon ? | Go to social-icon.svg  And del the fill=”#FFF”, the icon will appear black   </symbol>    <symbol xmlns="http://www.w3.org/2000/svg" viewBox="0 0 21 20" id="icon-instagram">      <path **fill="#FFF"** d="M10path>    </symbol>  Now add color in style.css that you have removed  .social-icon {    width: var(--size-600);  **fill: var(--clr-neutral-100);**  } | |
|  | How to reduce the height of icon ? | .social-icon {    width: var(--size-600);    fill: var(--clr-accent-300);  **height: min-content;**  }  Or .social-icon {    width: var(--size-600);    fill: var(--clr-neutral-100);  **aspect-ratio: 1;**  } | |
|  | To make 1st part of logo to green color ? | Go to logo.svg where you have added links from external svg website – go there and change the fill color to green  <svg width="0" height="0" class="hidden">    <symbol xmlns="http://www.w3.org/2000/svg" viewBox="0 0 146 24" id="logo">      <g fill="none" fill-rule="evenodd">        <path fill="#242D52" fill-rule="nonzero" d="M40.014"></path>  **<g fill="green">**          <path d="M2.786 10.286c1.538 0 2.785 1.279 2.785 2.857C5.571 14.72 4.324 16 2.786 16 1.247 16 0 14.72 0 13.143c0-1.578 1.247-2.857 2.786-2.857zM2.786 0C4.324 0 5.57 1.28 5.57 2.857c0 1.578-1.247 2.857-2.785 2.857C1.247 5.714 0 4.435 0 2.857 0 1.28 1.247 0 2.786 0zm10.028 0C14.353 0 15.6 1.28 15.6 2.857c0 1.578-1.247 2.857-2.786 2.857-1.538 0-2.785-1.279-2.785-2.857C10.029 1.28 11.276 0 12.814 0z"></path>          <ellipse cx="12.814" cy="13.257" opacity=".5" rx="2.786" ry="2.857"></ellipse>        </g>      </g>    </symbol>  </svg>  <svg>    <use xlink:href="#logo"></use>  </svg> | |
|  | To make 2st part of logo to green color ? | 1st go to logo.svg and add color to 2nd part as following as I have added variable of –logo-color  <svg width="0" height="0" class="hidden">    <symbol xmlns="http://www.w3.org/2000/svg" viewBox="0 0 146 24" id="logo">      <g fill="none" fill-rule="evenodd">  **<path fill="var(--logo-color)"** fill-rule="nonzero" d="M40.014  "></path>          <ellipse cx="12.814" cy="13.257" opacity=".5" rx="2.786" ry="2.857"></ellipse>        </g>      </g>    </symbol>  </svg>  In style.css:  .primary-footer {  **--logo-color: var(--clr-neutral-100);**  }  2ndly, go to html and make a class with name primary-header as following in footer  footer **class="primary-footer** | padding-block-700 bg-neutral-900 text-neutral-100">        <div class="container">  3rdly go to style and make function as  .primary-footer {  **--logo-color: var(--clr-neutral-100);**  } | |
|  | To change color of logo from html ? | <header class = "primary-header">        <div class = "container">          <div class = "nav-wrapper">            <a href="#">  **<svg style="--logo-color: purple">**                <use xlink:href="images/logo.svg#logo"></use>              </svg>            </a> | |
|  | To encircle the box where you are right now ? | .primary-footer **> \*** {  **outline: 2px solid red;**  }  .primary-footer **> \* > \* > \*** {  **outline: 2px solid red;**  } | |
|  | How to select 1st child of secton or class ? in footer  Only logo area in footer | .primary-footer-wrapper **> :nth-child(1)** {    grid-area: "logo-social";    background: red;  }  In html ?  <div class**="primary-footer-wrapper** even-columns">            <div>              <a href="#" aria-label="home" href="#">                <svg class="logo">                  <use xlink:href="images/logo.svg#logo"></use>                </svg>              </a>              <ul class="social-list" | |
|  | gap: var(--size-600) !important; | So, **gap: var(--size-600) !important;** means that the gap between grid items in the layout is set to the value of the custom property **--size-600**, and it's prioritized over any other conflicting styles due to the **!important** declaration. | |
|  | Remove the even column in footer to see easily footer ? | Remove even column      <footer class="**primary-footer** | padding-block-700 bg-neutral-900 text-neutral-100">        <div class="container">          <div class="primary-footer-wrapper **| even-column**"> | |
|  | .primary-footer-wrapper {  display: grid;  gap: var(--size-600);  grid-template-areas:  "form"  "nav"  "logo-social";  } | This arranges the primary footer wrapper class in sequence like form comes first than nav than logo  **grid-template-areas: "form" "nav" "logo-social";:**  This property defines the template for the grid areas within the .primary-footer-wrapper. It specifies how many rows and columns there are in the grid and what content should be placed in each area. In this case, there are three rows named "form", "nav", and "logo-social". | |
|  | .primary-footer-wrapper > :nth-child(1) {  grid-area: logo-social;    } | **.primary-footer-wrapper > :nth-child(1)**:   * This selector targets the first child element of **.primary-footer-wrapper**   **The elements of primary-footer-wrapper are:**  primary-footer-wrapper {    display: grid;    gap: var(--size-600);    grid-template-areas:      "form"      "nav"      "logo-social";  } | |
|  | @media (min-width: 50em) {  .primary-footer-wrapper {  **grid-template-areas:** "logo-social nav form";  }  } | When the size of window decrease, it must show 3 columsn of logo, nav and form | |
|  |  | create the name of classes before each logo, nav and form as following…  <div class="primary-footer-form">              <form action="">                <input type="email">                <button class="button"> | |
|  | How to reverse column or to invert the footer insta fb column situated below logo to move above logo ? | .primary-footer-logo-social {    grid-area: logo-social;    display: flex;  **flex-direction: column-reverse**;      } | |
|  | .primary-footer-logo-social {  **align-items: flex-start;**  justify-content: space-between;  gap: var(--size-700);  } | In footer, logo and insta,fb etc This CSS property specifies the alignment of the flex items along the cross axis. In this case, it sets the alignment to flex-start, meaning that the items will align to the start of the cross axis.  **justify-content: space-between;**:  This provide space between logo and fb, insta column | |
|  | .footer-nav {  columns: 2;  **gap: min(15vw, 2rem);**  }  Like those home, career navs in footer | So, **the gap between columns will be either 15% of the viewport width or 2 rems, whichever is smaller.** This ensures that the layout remains responsive and adapts to different screen sizes. | |
|  | How to change setting of form ? padding 1st is vertical than horiontal | input:where([type="text"], [type="email"]) {    border-radius: 100vw;    border: 0;    padding: var(--size-300) var(--size-400) ;  } | |
|  | Form and button, nav needs to be placed horizontally in footer | .**primary-footer form** {  **display: flex;**    gap: var(--size-400);  }  In html ?  <div class="primary-footer-form">              <form action="">                <input type="email">                <button class="button" data-shadow="none"> Go </button>                <p> Copyright 2024. All Rights Reserved. </p>              </form> | |
|  | After making class of form we placed outside the P – as we wanted the copy right in P should comes at bottom side |  | |
|  | To place a space btw form and copyright – first | Put grid than align content it will move down  .primary-footer-form {  grid-area: form;  **display: grid;**  **align-content: space-between**;    } | |
|  | The middle section numbered class needs to be numbered ? | .numbered-items {  **list-style: decimal !important**;  }  In HTML:  <ul class= **"numbered-items** | flow" role="list">                <li>                  <div class="flow" style="--flow-spacer: 1.5em">                    <h3 class="fs-600 fw-bold**"> Track company wide progress**                    </h3>                    <p data-width="wide">> · Outfit Font Family is an eye-catching sans serif font designed by font designer Rodrigo cda for use by automation specialists at outfit.</p>                  </div>                </li>                <li>                  <div class="flow" style="--flow-spacer: 1.5em">                    <h3 class="fs-600 fw-bold"> Advanced built in Reports </h3>                    <p data-width="wide"> · Outfit Font Family is an eye-catching sans serif font designed by font designer Rodrigo Fuenzalida for use by automation specialists at outfit.</p>                  </div>                </li> | |
|  | To change color of numbers and make it bold ? | Use marker for that purpose  .**numbered-items ::marker** {  **color: red;**  **font-weight: bold;**  } | |
|  | .numbered-items {  counter-reset: count;  }  .numbered-items li {  counter-increment: count;  }  .numbered-items li::before {  content: counter(count);  } | Now, regarding the line **content: counter(count);**:   * **content**: This property is used to specify the content that will be generated by the **::before** or **::after** pseudo-elements. * **counter(count)**: Here, **counter()** is used to retrieve the value of a counter named **count**. This means that whatever value the counter **count** holds will be inserted as the content before each list item.   counter(): The counter() function is used to retrieve the current value of a counter. It takes the name of the counter as its argument.  counter-increment: This property is used to increment the value of a counter. You can specify the name of the counter you want to increment.  counter-reset: This property is used to reset the value of a counter to a specific value. | |
|  | To put no along with “text” in center section ? To cope with problem that Para don’t come along with it ? | In HTML: we have class of items and than Ul  <ul **class= "numbered-items** | flow" role="list">  <li>  **<div** class="flow" style="--flow-spacer: 1.5em">  <h3 class="fs-600 fw-bold"> **Track company wide progress**  </h3>  <p data-width="wide">> · Outfit Font Family is an eye-catching sans serif font designed by font designer Rodrigo cda for use by automation specialists at outfit.</p>  **</div>**  </li>  In css style ? write class item along with div and use 1fr 1fr instead of using min-content along with it  .numbered-items {    counter-reset: count;  }  .numbered-items li {    counter-increment: count;  }  **.numbered-items div {**  **display: grid;**  **grid-template-columns: 1fr 1fr;**  **}**  .numbered-items div::before {    content: "0" counter(count);    background-color: var(--clr-accent-400);    color: var(--clr-neutral-100);    font-weight: var(--fw-bold);    padding: var(--size-100) var(--size-500);    border-radius: 100vw;  } | |
|  | .numbered-items p {    grid-column: 1/-1;  } | * + **1** refers to the start line of the grid track.   + **-1** refers to the end line of the grid track, counting from the end of the grid area. Negative values in grid-column and grid-row indicate counting from the end of the grid area.   Therefore, **grid-column: 1/-1;** specifies that the **p** elements within **.numbered-items** will span from the first column to the last column of the grid layout. This effectively makes them occupy the full width of their container, regardless of the number of columns defined in the grid. | |
|  | How to align the heading next to number at center horizontally ? and how to make gap btw them ? | .numbered-items div {    display: grid;  **align-items: center;**  **column-gap: var(--size-400);**    grid-template-columns: min-content 1fr 1fr;  } | |
|  | .numbered-items p {  grid-column: 2 / -1;  } | 1. **grid-column: 2 / -1;**: This CSS declaration sets the grid-column property of the targeted **<p>** elements. In grid layout, grid-column determines the range of columns a grid item occupies.    * **2** refers to the start line of the grid track, indicating that the **<p>** elements will start from the second column.    * **-1** refers to the end line of the grid track, counting from the end of the grid area. Negative values in grid-column and grid-row indicate counting from the end of the grid area. | |
|  | .numbered-items\_\_title::before {  **content: ".";**  position: absolute;  **inset: 0;**  grid-column: 1 / -1;  grid-row: 1/2;  **background: red;**  **z-index: -1;**  } | Content if has dot it will place in start of title  Inset with 0 will place whole section with red color  Grid column mean it will start from column 1 and end at end of column like the shadow color along with number in horizontal direction  z-index= -1 will write on horizontal line of red  In this case, setting a negative **z-index** ensures that the pseudo-element is positioned behind other elements on the page. This might be useful if you want to create a layered effect where the content of the pseudo-element appears behind the content of other elements. However, it's important to note that the **z-index** property only works on positioned elements (elements with **position: relative**, **position: absolute**, **position: fixed**, or **position: sticky**).  Top of Form  The **inset** CSS property is a shorthand property that allows you to set the values for the top, right, bottom, and left positions of a positioned element all at once. It is commonly used in conjunction with the **position** property, typically with **position: absolute** or **position: fixed**.  Here's how **inset** works:   * If you provide one value, it applies to all four sides (top, right, bottom, left). * If you provide two values, the first value applies to top and bottom, and the second value applies to left and right. * If you provide three values, the first value applies to top, the second value applies to left and right, and the third value applies to bottom. * If you provide four values, they apply to top, right, bottom, and left respectively.   Pseudo-elements in CSS, such as **::before** and **::after**, allow you to insert content into the HTML document dynamically without modifying the actual HTML markup. They are useful for adding decorative elements, styling generated content, or implementing certain design effects. Here are some reasons why you might use the **::before** pseudo-element: | |
|  | .numbered-items\_\_title::before {  content: ".";  position: absolute;  **inset: 0 -100vw 0 0;**  grid-column: 1 / -1;  grid-row: 1/2;  background: orange;  border-radius: 100vw;  z-index: -2;  } | In CSS, the inset property is a shorthand property for specifying the top, right, bottom, and left offsets of absolutely positioned elements. It's similar to top, right, bottom, and left properties combined into one.  Grid row ? In CSS Grid layout, the **grid-row** property specifies the placement of grid items along the grid's vertical axis. It defines the **start and end lines of the grid area** that the element occupies. | |
|  | .numbered-items div::before {  content: "0" counter(count);  background-color: var(--clr-accent-400);  color: var(--clr-neutral-100);  font-weight: var(--fw-bold);  **padding: var(--size-300) var(--size-500);**  border-radius: 100vw;  } | To adjust the number icon with acc to shadow horzontal | |
|  | .numbered-items div::before {  content: "0" counter(count);  background-color: var(--clr-accent-400);  color: var(--clr-neutral-100);  font-weight: var(--fw-bold);  **padding: 0 var(--size-500);**  **height: 50px;**  border-radius: 100vw;  } | To increase decrease height of number by padding 1st 0 | |
|  | If no 2 is not mod than ? | Add display flx and align-items: center  .numbered-items div::before {  **display: flex;**  **align-items: center;**    content: "0" counter(count);    background-color: var(--clr-accent-400);    color: var(--clr-neutral-100);    font-weight: var(--fw-bold);    padding: 0 var(--size-500);    height: 50px;    border-radius: 100vw;  } | |
|  | .**numbered-items div::after** {    content: ".";  display: block;  position: absolute;  inset: 0 -100vw 0 0;  **grid-column: 1 / -1;**  **grid-row: 1/2;**  background: orange;  width: 100%;  /\*height: 40px;\*/  border-radius: 100vw;  z-index: -1;  } | We change the class from title to div::after  We change the column from ½ to 1/-2 as the shadow line plus numbered icon was not appearing  In CSS, .numbered-items div::after is a selector that targets pseudo-elements that are generated and inserted after each <div> element within elements that have the class numbered-items. Here's what each part of the selector means: | |
|  | To put line height space, not go close to boundaries ? | .numbered-items\_\_title {    grid-column: 2 / 3;    grid-row: 1/2;  **line-height: 1;**    ;  } | |
|  | How to control height of both function in one common function ? | .**numbered-items div::before,**  **.numbered-items div::after** {    height: 50px;  }  The CSS code you provided targets pseudo-elements generated before (**::before**) and after (**::after**) each **<div>** element within elements having the class **.numbered-items**.  Here's what it does:   * **.numbered-items div::before** targets the pseudo-elements generated before each **<div>** element within elements with the class **.numbered-items**. * **.numbered-items div::after** targets the pseudo-elements generated after each **<div>** element within elements with the class **.numbered-items**.   Both rules set the **height** property of these pseudo-elements to **50px**, meaning that they will have a height of 50 pixels. | |
|  | border-radius: 100vw 0 0 100vw; | he CSS property **border-radius** is used to create rounded corners for an element. The value **100vw 0 0 100vw** you provided represents four values, each specifying the radius for a different corner of the element in clockwise order starting from the top-left corner.   * **100vw**: This value specifies that the top-left and bottom-right corners should have a radius equal to 100% of the viewport width. This would create a rounded corner that extends all the way from the top-left corner to the bottom-right corner of the viewport. * **0**: This value specifies that the top-right corner should have a radius of 0, resulting in a sharp corner. * **0**: This value specifies that the bottom-right corner should have a radius of 0, resulting in a sharp corner. * **100vw**: This value specifies that the bottom-left corner should have a radius equal to 100% of the viewport width, similar to the top-left corner.   So, overall, **border-radius: 100vw 0 0 100vw;** creates a shape that looks like a horizontally elongated ellipse with rounded ends, spanning the entire width of the viewport and having sharp corners on the top-right and bottom-right. | |
|  | How to put box shadow in place of heading – light orange color?  How to put border radius and along with color? | @media (max-width: 30em)    {    .numbered-items div::after {  **--bg: var(--clr-accent-100);**      content: ".";      display: block;      grid-column: 1 / -1;      grid-row: 1/2;      background: var(--bg);  **border-radius: 100vw 0 0 100vw**;      z-index: -1;  **box-shadow:  5em 0 0 var(--bg);**      }    } | |
|  | Width is fit content ?? | In css ? fit content lies btw max and min content related to shrink os screen  .numbered-items {  counter-reset: count;  margin-inline: auto;  **width: fit-content;**  }  In html ?  <ul **class= "numbered-items** | flow" role="list">  <li>  <div class="flow" style="--flow-spacer: 1.5em">  <h3 class="numbered-items\_\_title fs-600 fw-bold"> Track company wide progress  </h3>  <p class="numbered-items\_\_body" data-width="wide"> Outfit Font Family is an eye-catching sans serif font designed by font designer Rodrigo cda for use by automation specialists at outfit.</p>  </div>  </li> | |
|  | How to align heading to center when at min screen ? | In html ?  <h2 class**="text-center-sm-only** fs-secondary-heading fw-bold">  Bring everyone together to build better products.  In css ?    **.text-center {**  **text-align: center;**  }  @media **(min-width**: 50em) {  **.text-center-sm-only** {  text-align: center;  }  } | |
|  | How to put p at center? Conflict is ooccuring | In HTML ?  .text-center {  text-align: center;  }  **.text-center p,**  **.even-columns p {**  **margin-inline: auto ;**    **}**  @media (max-width: 50em) {  .text-center-sm-only {  text-align: center;  }  .**text-center-sm-only p {**  **margin-inline: auto ;**    }  }  In css ?  <div class="container">        <div class**="even-columns">**          <div class="flow">            <h2 class="**text-center-sm-only** fs-secondary-heading fw-bold">              Bring everyone together to build better products.            </h2>              <**p** data-width="wide">                Manage make it simple for everything and you should the real essence of working hard but not smart  **</p>**            <div>  In your CSS code, you have defined two classes: **.text-center** and **.text-center-sm-only**. The **.text-center** class applies the **text-align: center;** property to its elements, which should center-align the text within those elements. Similarly, **.text-center p** targets **p** elements within elements with the class **.text-center** and sets their **margin-inline** property to **auto**, which should center them horizontally within their container.  However, in your HTML code, the paragraph (**<p>**) element is not directly wrapped in an element with the class **.text-center**. Instead, it's inside an element with the class **.even-columns**. Therefore, the styles defined for **.text-center p** won't apply to this paragraph because it doesn't meet the CSS selector specificity.  To fix this, you need to either add the **.text-center** class directly to the container element that wraps the paragraph or adjust your CSS to target the paragraph based on its parent class **.even-columns**. Here's how you can adjust your CSS: | |
|  | How to add margin-bottom-700 size in mid section top heading, to place space underneath it  For bubblle | In HTML?  <div class="even-columns">  <div class="flow text-center-sm-only **margin-bottom-700**" style="--flow-spacer: 2rem">  <h2 class="fs-secondary-heading fw-bold" >  Bring everyone together to build better products.  </h2>  <p data-width="wide">  Manage make it simple for everything and you should the real essence of working hard but not smart  </p>  </div>  In css?  .**margin-bottom-700** {  **margin-block-end:** var(--size-700);  } | |
|  | **text-center-sm-only** ?  note: div should be btw H and P | In HTML?  <div class="even-columns">  <**div** class="flow **text-center-sm-only** margin-bottom-700" style="--flow-spacer: 2rem">  <h2 class="fs-secondary-heading fw-bold" >  Bring everyone together to build better products.  </h2>  <p data-width="wide">  Manage make it simple for everything and you should the real essence of working hard but not smart  </p>  </**div**>  In CSS ?  @media (max-width: 50em) {  .**text-center-sm-only** {  text-align: center;  }  .**text-center-sm-only p** {  margin-inline: auto ;  }  } | |
|  | How to add slider ? | Go to website a11y and copy links and pasete it in html, also write https before //:  <link rel="stylesheet" href="**http://**unpkg.com/a11y-slider@latest/dist/a11y-slider.css" />  <!-- End of <body> -->  <script src="//unpkg.com/a11y-slider@latest/dist/a11y-slider.js"></script>  Add these lines from website  .slider {  display: flex;  }  .slider > \* {  width: 100%;  flex: 0 0 auto;  } | |
|  | Slider was not showing ? | Add this in js:  const slider = new A11YSlider(document.querySelector('.slider'), {  adaptiveHeight: true,  dots: false  });  These lines of code initialize a new instance of an accessibility-friendly slider component using a JavaScript library called A11YSlider. Let's break down what each part of the code does:  const slider = new A11YSlider(...): This line declares a constant variable named slider and assigns it the result of creating a new instance of the A11YSlider class. The A11YSlider class is likely defined within the A11YSlider library or module.  document.querySelector('.slider'): This part of the code uses the document.querySelector() method to select an HTML element with the class name "slider". This is typically a container element that will hold the slider component.  adaptiveHeight: true: This option configures the slider to adjust its height dynamically based on the height of the currently active slide. When set to true, the slider will adapt its height to fit the content of the current slide.  dots: false: This option determines whether navigation dots will be displayed on the slider. When set to false, navigation dots are disabled, meaning users won't see visual indicators for each slide. This can be useful for customizing the appearance of the slider interface.  In summary, these lines of code initialize a slider component with accessibility features, such as adaptive height adjustment, and configure it to disable navigation dots. | |
|  | .slider {  display: flex;  list-style: none;  } | list-style: none; so that we cannot see dot | |
|  | How to add image btw li?  Hpw to add heading ? and para ?  How to aligntext center ? | <section class="**carousel** | padding-block-700 **text-center**">    <**h2** class="fs-secondary-heading fw-bold">  What they have said  </h2>  <ul **class="slider">**  <li>  <**img class="icon-close" src="images/avatar-ali.png" alt="">**  <h3 class="fw-bold"> Ali Bravo </h3>  **<p data-width="wise"**> QuillBot is a free and easy-to-use tool that rephrases text in different modes and styles. It helps you improve your fluency, vocabulary </p>  </li>  <li>2</li>  <li>3</li>  <li>4</li>  </ul> | |
|  | How to reduce size of image of slider ? | .slider img {  width: var(--size-800);  position: absolute;  } | |
|  | The footer segment is interfering the or covering the feedback section ? | padding: var(--size-900) var(--size-400) var(--size-400);  The **top** padding is var(--size-900) (e.g., 20px).  **The right and left** paddings are both var(--size-400) (e.g., 10px).  **The bottom pa**dding is also var(--size-400) (e.g., 10px).  The actual pixel values would depend on the specific values assigned to --size-900 and --size-400. | |
|  | .slider > \* means ? | .slider: This is a class selector targeting elements with the class name "slider".  **>:** This is the child combinator selector, which **selects only the direct children of the preceding element.**  **\*:** This is a universal selector, **which matches any element type.**  So, .slider > \* selects all direct children of elements with the class .slider, regardless of their element type. In the context of your CSS code snippet, it's being used to style the direct children of elements with the class .slider, applying pffadding, positioning, width, flex properties, background color, and border radius to them. | |
|  | I want to move picture up and down and adjust? | **In css ?**  .slider img {  width: var(--size-800);  position: absolute;  inset: 0;  margin-inline: auto;  top: 5%;  **transform: translateY(0%);**  }  **translateY(): This function moves the element vertically.** Positive values move the element downward, while negative values move it upward.  0%: This specifies the distance of the translation as a percentage of the element's own height. Here, **0% means that the element will be translated by 0% of its height**, effectively not moving it vertically at all.  So, transform**: translateY(0%); essentially means that the element will not be moved vertically from its original position**. It's a way to explicitly set the vertical translation to zero, ensuring that the element remains in its current vertical position. | |
|  | I have made a whole section of feedback but still the gap is near, gap mean breaker ? | .slider > \* {  padding: var(--size-800) var(--size-400) var(--size-400);  position: relative;  **width: 400px;**  flex: 0 0 auto;  background: var(--clr-neutral-200);  border-radius: var(--size-600);    }  Use width 400px instead of 100% | |
|  | In feedback section, how to put box background only on content ? like para and heading not on pic ? | **1st select all para and heading using ALT key**  **2nd use key “ctrl + Shift + P”, and write emmet, click on wrap with abbreaviation**  **Thwn write .slider-content**  **Go to css slider > \* class and remove background and border radius and make new class and copt that in it**  **.slider-content {**  **background: var(--clr-neutral-200);**  **border-radius: var(--size-600);**  **}** | |
|  | After using emmet, what happen ? html chages ? | **<ul class="slider">**  **<li>**  **<img src="images/avatar-ali.png" alt="">**  **<div class="slider-content">**  **<h3 class="fw-bold"> Ali Bravo </h3>**  **<p data-width="wise"> QuillBot is a free and easy-to-use tool that rephrases text in different modes and styles. It helps you improve your fluency, vocabulary </p>**  **</div>**  </li>  <li>  **<img src="images/avatar-richard.png" alt="">**  **<div class="slider-content">**  <h3 class="fw-bold"> Richard </h3>  <p data-width="wise"> QuillBot is a free and easy-to-use tool that rephrases text in different modes and styles. It helps you improve your fluency, vocabulary </p>  **</div>**    </li>  <li>  <img src="images/avatar-anisha.png" alt="">  <div class="slider-content">  <h3 class="fw-bold"> Anisha </h3>  <p data-width="wise"> QuillBot is a free and easy-to-use tool that rephrases text in different modes and styles. It helps you improve your fluency, vocabulary </p>  </div>  </li>  <li>  <img src="images/avatar-shanai.png" alt="">  <div class="slider-content">  <h3 class="fw-bold"> Shanai </h3>  <p data-width="wise"> QuillBot is a free and easy-to-use tool that rephrases text in different modes and styles. It helps you improve your fluency, vocabulary </p>  </div>  </li>  </ul> | |
|  | To bring pic at mid of top line of background box ? | .slider img {  width: var(--size-800);  position: relative;    margin-inline: auto;  **top: calc(var(--size-800) / 2);** | |
|  | To add slider … feature or to make responsive the website ? | **Go to a11y slider website and see the content of …slide copy the code in to JS script ,the code in website was ?**  **responsive: {**  **768: {**  **slidesToShow: 2,**  **arrows: false**  **},**  **960: {**  **disable: true // slider disabled 960px to 1279px**  **},**  **1280: {**  **disable: false,**  **slidesToShow: 4,**  **dots: true // dots enabled 1280px and up**  **}**  **}**  **});**  **Now copy this site code in JS script ?**  const slider = new A11YSlider(document.querySelector('.slider'), {  adaptiveHeight: false,  dots: true  **responsive: {**    **800: {**  **dots: false, // dots enabled 1280px and up**  **},**  **},**  }); | |
|  | Add miin width  50em x 16 = px | **@media(min-width:30em) {**  **.slider > \* {**  **width: 50%;**  **}**  **}**  **@media(min-width:50em) {**  **.slider > \* {**  **width: 35%;**  **}**  **}** | |
|  | To add margin block in order to broaden the gap of slider section to footer ? | **.a11y-slider-container {**  **margin-block: var(--size-700);**  **}**  **A11y is get by inspect in which whole section got catched** | |
|  | To add dots in slider section ? | **Go to a11y site, and inspect to find sliders in form of dots**  **Copy them to style.css**  **a11y-slider-dots li {**  **display: block;**  **width: 12px;**  **height: 12px;**  **padding: 0;**  **margin: 0 6px;**  **color: #fff;**  **}**  **.a11y-slider-dots li button {**  **display: block;**  **font-size: 0;**  **text-indent: -9999px;**  **border: 1px solid !important;**  **width: 100%;**  **height: 100%;**  **border-radius: 100%;**  **}** | |
|  | To change the color of dots ? | .a11y-slider-dots li button {  display: block;  font-size: 0;  text-indent: -9999px;  width: 100%;  height: 100%;  border-radius: 100%;  **border: 1px solid var(--clr-accent-400);**  }  Through inspect verify the border icon | |
|  | How to make background of dots transparent ? | .a11y-slider-dots li button {  display: block;  font-size: 0;  text-indent: -9999px;  width: 100%;  height: 100%;  border-radius: 100%;  **border: 1px solid var(--clr-accent-400);**  **background: transparent;**  } | |
|  | How to put dots on center ? | .a11y-slider-dots {  display: flex;  gap: var(--size-300);  **justify-content: center;**  } | |
|  | How to change background color when dot is selected or active ? | **Go to inspect and find the dot command and than write following.**  **a11y-slider-dots li button.active{**  **background: var(--clr-accent-400);**  **}** | |
|  | To remove or add prev and next button in feedback section ? | const slider = new A11YSlider(document.querySelector('.slider'), {  adaptiveHeight: false,  dots: true,  centerMode: true,  **arrows: false,**  responsive: {    480: {  dots: false, // dots enabled 1280px and up  **arrows: true,**  },  },  }); | |
|  | To hide prev button of slider ? | text-indent: -9999px;: This property is often used to visually hide text while keeping it accessible to screen readers and other assistive technologies. By setting the text indent to a large negative value, such as -9999px, the text is effectively pushed far off-screen. This technique is commonly used for elements like buttons or links that have text content but need to display custom background images or icons instead.  position: relative;: This property establishes the positioning context for the element. Setting it to relative means that the element is positioned relative to its normal position in the document flow. This property is often used in conjunction with other positioning properties like top, bottom, left, and right to precisely control the element's placement.  **.a11y-slider-prev {**  **text-indent: -9999px;**  **position: relative;**  **}** | |
|  | To make margin equivalent at right side ? at min 30em | @media(min-width:50em) {  .slider > \* {  width: 35%;  }  .**a11y-slider-container {**  **margin-inline: 0;**  }  } | |
|  | .a11y-slider-container {  margin-block: var(--size-700);  margin-inline: var(--size-400);    } | **This CSS snippet targets an element with the class .a11y-slider-container and applies margins to it. Let's break down each property:**  **margin-block: var(--size-700);: This property sets the margin in the block direction, which is typically the vertical direction in most documents. The value is defined using a custom property --size-700. The actual size of the margin would depend on the value assigned to --size-700 elsewhere in your CSS code. For example, if --size-700 is defined as 20px, then this would create a vertical margin of 20px for the .a11y-slider-container element.**  **margin-inline: var(--size-400);: This property sets the margin in the inline direction, which is typically the horizontal direction in most documents. Again, the value is defined using a custom property --size-400, whose actual size would be defined elsewhere in your CSS code. For example, if --size-400 is defined as 10px, then this would create a horizontal margin of 10px for the .a11y-slider-container element.**  **In summary, the .a11y-slider-container element would have margins applied to it, with the specific margin sizes determined by the values assigned to the custom properties --size-700 and --size-400.** | |
|  | How to put border around column ? | **<div style="border: 1px solid red**;">              <img src="images/illustration-intro.svg" alt="manage">            </div> | |
|  |  | In html make class of :  <div class="**image-with-blob-bg**">  <img src=s"images/illustration-intro.svg" alt="manage">  </div>  In css make class as :  .image-with-blob-bg {  position: relative;  }  .image-with-blob-bg::before {  position: absolute;  content: "";  left:0;  top:0;  width: 100%;s  aspect-ratio: 1;  background-image: url("images/bg-tablet-pattern.svg");  background-position: bottom left;  }  When you put aspect ratio in css, the bubble pic appear  To place it on right direction, we put background positions | |
|  | Before and afters | **::before / ::after. Sara Cope on Sep 6, 2011 (Updated on Sep 30, 2022 ) The ::before and ::after pseudo-elements in CSS allows you to insert content onto a page without it needing to be in the HTML.** | |
|  | To make bottom left corner of svg pics | .image-with-blob-bg::before {  content: "";  position: absolute;  left:0;  top:0;  width: 400px;  aspect-ratio: 1;  z-index: -1;  background-image: url("images/bg-tablet-pattern.svg");  background-repeat: no-repeat;  **background-position: bottom left;**  border: 1px solid red;  }  background-repeat: no-repeat; to not to repeat the background  top: 0 right:0s to place at top right side s | |
|  | aspect-ratio | **The aspect-ratio property is a newer CSS property that allows you to control the aspect ratio of an element, which is the ratio of its width to its height.**  **In your example, aspect-ratio: 1/1.3; sets the aspect ratio of the pseudo-element to 1:1.3. This means that for every 1 unit of width, the height will be 1.3 units.**  **.**image-with-blob-bg::before {  content: "";  position: absolute;  right:0;  top:0;  width: 400px;  **aspect-ratio: 1/1.3;**  z-index: -1;  background-image: url("images/bg-tablet-pattern.svg");  background-repeat: no-repeat;  background-position: bottom left;  border: 1px solid red;  } | |
|  | Order is used to place item or image on 1st 2nd grid, if used -1 order than it will place at 1st left side | **.**image-with-blob-bg {  **order: -1;**  } | |
|  | To put text at center, of main 1st section ? | <section class="hero | **text-center-sm-only** | padding-block-900" >    <div class="container"> | |
|  | Margin-inline | margin-inline resolves this issue by representing the horizontal margin, regardless of the writing mode. When you use margin-inline: auto;, it applies an equal amount of margin space on both sides of the element, effectively centering it horizontally within its containing block. | |
|  | To place the picture of illustration on centre | In html ?  <div class="hero\_\_image">  **<img class="mx-auto" src="images/illustration-intro.svg"** alt="manage">  </div>  In css ?  **.mx-auto** {  margin-inline: auto;  } | |
|  | If you want your svg image to not move up and down, and get stagnated and move left right how ? buble image ? | .hero\_\_image::before {  content: "";  position: absolute;  right:0;  top:0;  width: 87%;  **max-height: 83vh;**  aspect-ratio: 1/1.5;  z-index: -1;  background-image: url("images/bg-tablet-pattern.svg");  background-repeat: no-repeat;  background-position: bottom left;  border: 1px solid red;  } | |
|  | Top right nav button list background is transparent? | @media (max-width: 50em) {  .primary-navigation {  display: none;  padding: var(--size-700);  **z-index: 999;**  max-widsth: 25rem;  margin-left: auto;  position: fixed;  inset: 7rem var(--size-400) auto;  background: var(--clr-neutral-100);  border-radius: var(--size-100);  box-shadow: 0 0 0 0.75em rgb(0, 0, 0, 0.05);  } | |
|  | To place top right nav button list when opened at center ? | @media (max-width: 50em) {  .primary-navigation {  display: none;  padding: var(--size-700);  z-index: 999;  max-width: 25rem;  **margin-inline: auto;**  position: fixed;  inset: 7rem var(--size-400) auto;  background: var(--clr-neutral-100);  border-radius: var(--size-100);  box-shadow: 0 0 0 0.75em rgb(0, 0, 0, 0.05);  } | |
|  | To make menue button static in such way that it will show icon all the time ? add url in both  Fix the width of menu icon | .primary-navigation[data-visible] {  display: block;    }    .mobile-nav-toggle {  display: block;  position: fixed;  top: var(--size-600);  right: var(--size-400);  cursor: pointer;  padding: 0.5em;  background: transparent;  border:0em;  **width:2em;**  **aspect-ratio:1;**  **background-image: url("images/icon-close.svg");**  background-size: contain;  }  **.mobile-nav-toggle[aria-expanded="true"] {**  **background-image: url("images/icon-close.svg");**    **}** | |
|  | To make another bubble at center section at right side  Oom is used to small the size of svg bubble | In html ?  <section **class="sales\_point** | padding-block-900">    <div class="container">  <div class="even-columns">  <div class="flow text-center-sm-only margin-bottom-700" style="--flow-spacer: 2rem">  <h2 class="fs-secondary-heading fw-bold" >  Bring everyone together to build better products.  </h2>  <p data-width="wide">  Manage make it simple for everything and you should the real essence of working hard but not smart  </p>  </div>  <div>  In css ?  .sales\_point {  position: relative;  }  .**sales\_point::before** {  **content: url("images/bg-tablet-pattern.svg");**  position: absolute;  z-index: -1;  **zoom: 25%;**  **left:75%;**  **bottom:100%;**  border: 2px solid red;  width: 200px;  aspect-ratio: 1/1.5;        } | |
|  | html,  body {  height: 100%;  overflow-x:hidden;  } | ?????? | |
|  | Add one bubble when at mobile size and one at 47em ?  Change the name of class ? | In html?  <section class="**sales\_point** | padding-block-900">    <div class="container">  <div class="even-columns">  <div class="sales\_**point\_\_blog** | flow text-center-sm-only margin-bottom-700" style="--flow-spacer: 2rem">  <h2 class="fs-secondary-heading fw-bold" >  Bring everyone together to build better products.  </h2>  <p data-width="wide">  Manage make it simple for everything and you should the real essence of working hard but not smart  </p>  In css ?  .sales\_**point\_\_blog** {  position: relative;  }  .sales\_point**\_\_blog**::before {  content: url("images/bg-tablet-pattern.svg");  position: absolute;  z-index: -1;  zoom: 40%;  left:75%;  bottom:100%;  width: 200px;  aspect-ratio: 1/1.5;  }  **@media (min-width: 47em)** {  .sales\_**point\_\_blog**::before {  z-index: -1;  zoom: 80%;  left:-50%;  top:45%;  width: 200px;  aspect-ratio: 1/1.5;    }  } | |
|  | How to make space before top of LAST section ? | In html ?  <section class="cta | **margin-top-900** | padding-block-700 bg-accent-400 text-neutral-100">    <div class="container">  <div class="even-columns vertical-align-center">  <d  In css ?  .margin-top-900 {  margin-block-start: var(--size-900);  } | |
|  |  | In html:  <footer class="primary-footer | padding-block-700 bg-neutral-900 text-neutral-100">  <div class="container">  <div class="primary-footer-wrapper">    <a class="primary-footer-logo" href="#" aria-label="home" href="#">  <svg class="logo">  <use xlink:href="images/logo.svg#logo"></use>  </svg>  </a>  <ul class="social-list" role="list" aria-label="Social links">    <li> <a aria-label="facebook" href="#"> </a>  <svg class="social-icon">  <use xlink:href="images/social-icons.svg#icon-facebook"></use>  </svg>  </li>  **In css:**  .primary-footer-wrapper {  display: grid;  gap: var(--size-600);  **grid-template-areas:**  **"form"**  **"nav"**  **"social"**  **"logo"**  **"copyright";**  }  **.primary-footer-logo {**  **grid-area:logo;**  **}**  @media (min-width: 50em) {  .primary-footer-wrapper {  grid-template-areas: "logo-social nav form";  }  /\* .primary-footer-logo-social {  align-items: flex-start;  justify-content: space-between;  gap: var(--size-700);  }\*/  }  In CSS, the .primary-footer-logo selector followed by the declaration grid-area: logo; defines the layout of an element with the class primary-footer-logo within a CSS Grid layout.  grid-area: logo; assigns the grid area named logo to the element. This means that the element will occupy the grid cells defined for the logo area within the grid container it belongs to.  In a CSS Grid layout, you define named areas within the grid using the grid-template-areas property on the container element. Then, you assign these named areas to child elements using the grid-area property, as demonstrated in your example. This allows for a more semantic and flexible way to lay out elements within a grid. | |
|  | Put all names in footer with primary-footer and double underscore with name ?  Do the same in css | **In css ?**  <footer class**="primary-footer** | padding-block-700 bg-neutral-900 text-neutral-100">  <div class="container">  <div class="**primary-footer\_\_wrapper**">    <a class="primary-footer\_\_logo" href="#" aria-label="home" href="#">  <svg class="logo">  <use xlink:href="images/logo.svg#logo"></use>  </svg>  </a>  <ul class="social-list" role="list" aria-label="Social links">    <li> <a aria-label="facebook" href="#"> </a>  <svg class="social-icon">  <use xlink:href="images/social-icons.svg#icon-facebook"></use>  </svg>  </li>  <li> <a aria-label="Youtube" href="#"> </a>  <svg class="social-icon">  <use xlink:href="images/social-icons.svg#icon-youtube"></use>  </svg>  </li>  <li> <a aria-label="Twitter" href="#"> </a>  <svg class="social-icon">  <use xlink:href="images/social-icons.svg#icon-twitter"></use>  </svg>  </li>  <li> <a aria-label="pinterest" href="#"> </a>  <svg class="social-icon">  <use xlink:href="images/social-icons.svg#icon-pinterest"></use>  </svg>  </li>  <li> <a aria-label="Instagram" href="#"> </a>  <svg class="social-icon">  <use xlink:href="images/social-icons.svg#icon-instagram"></use>  </svg>  </li>  </ul>    <div class="primary-footer\_\_nav">  <nav class="footer-nav">    <ul class="flow" style="--flow-spacer: 1em" aria-label="Footer" role="list">  <li> <a href="#"> Home </a> </li>  <li> <a href="#"> Pricing </a> </li>  <li> <a href="#"> Products </a> </li>  <li> <a href="#"> About Us </a> </li>  <li> <a href="#"> Careers </a> </li>  <li> <a href="#"> Community </a> </li>  <li> <a href="#"> Privacy Policy </a> </li>  </ul>    </nav>  </div>  {% comment %} <div class="primary-footer-form"> {% endcomment %}  <form class="**primary-footer\_\_form**" action="">  <input type="email">  <button class="button" data-shadow="none"> Go </button>  </form>  <p class="**primary-footer\_\_copyright**"> Copyright 2024. All Rights Reserved. </p>  {% comment %} </div> {% endcomment %}    </div>      </div>      </footer>  <div id="app"></div>  <script type="module" src="/src/main.ts"></script>  </body>  </html>  **In css ?** | |
|  | To put all content at center ? footer | **In html ?**  <footer class="primary-footer | padding-block-700 bg-neutral-900 text-neutral-100">  <div class="container">  <div class="**primary-footer\_\_wrapper">**    **In css ?**  .**primary-footer\_\_wrapper** {  display: grid;  gap: var(--size-600);  **justify-content: center;**  grid-template-areas:  "form"  "nav"  "social"  "logo"  "copyright";  } | |
|  | To put logo at footer at center ? | **In html ?**  <footer class="primary-footer | padding-block-700 bg-neutral-900 text-neutral-100">  <div class="container">  <div class="primary-footer\_\_wrapper">    <a class="primary-footer\_\_logo" href="#" aria-label="home" href="#">  **<svg class="logo | mx-auto**">  <use xlink:href="images/logo.svg#logo"></use>  </svg>  </a>  **In css ?**  **.mx-auto {**  **margin-inline: auto;**  **}** | |
|  | Social icon size in footer is size--600 | | |
|  | To close the gap btw nav in footer | .primary-footer\_\_nav {  grid-area: nav;  border: 2px solid red;  margin-inline: auto;    } | |
|  |  | .footer-nav {  **columns: 2;**  **gap: clamp(4rem, 10vw, 7rem);**  }  This code is styling a .footer-nav element, presumably intended for a footer navigation section. Let's break down what each property does:  columns: 2;: This property is setting the number of columns in which the content of .footer-nav will be displayed. In this case, it specifies that the content should be displayed in two columns.  gap: clamp(var(--size-200), 30vw, var(--size-900));: This property sets the gap (or space) between the columns in the grid layout. It uses the clamp() function to set a flexible gap size based on a minimum, preferred, and maximum value.  var(--size-200) represents the minimum gap size. It's likely a custom CSS variable (--size-200) defined elsewhere in the stylesheet, representing a specific size, probably in pixels.  30vw represents the preferred gap size. It uses the viewport width (vw) unit, which is relative to the width of the viewport. This means that the gap will be 30% of the viewport width when the viewport is at its full width.  var(--size-900) represents the maximum gap size. Similar to var(--size-200), it's likely a custom CSS variable representing a maximum size, probably in pixels.  So, in summary, this CSS rule is styling a footer navigation section to display its content in two columns with a flexible gap size between the columns, which varies based on the viewport width within a specified range. | |
|  | To place text at right footerside or center | .primary-footer\_\_copyright {  text-align: right;  } | |

* 1. Data overlay in java
* 2- p:not([class]){
* opacity: 0.9;
* max-width: 32ch;
* ;
* }

3- Before after ? z-index ? aspect ratio ?

* 1. Section CTA ma button is not moving on right side using justify ? why

1. .justify-self-end {
2. justify-self:end;
3. }

4- line 180 html, ma yeh “md”, daalnay sa mid a kesy a agyaya ?

"justify-self-end-md"

5- How to give gap btw two lines ? like form and P